Overview of the "*SiriusQuality-BioMa-Phenology-Component"* folder

1. Content of the "SiriusQuality-BioMa-Phenology-Component" folder

   The folder "SiriusQuality-BioMa-Phenology-Component" contains:

   - The source code of the *SQ-Phenology BioMa* Component (*SiriusQuality-PhenologyComponent* folder), see *SQ2_Phenology_component* document
   - The BioMa dll which are mandatory to use the component (*BioMa-DLL* folder)
   - A console application which provides an example for the use of the component (*SiriusQuality-PhenologyConsole* folder)
   - Unit tests for the component (*UnitTestPhenology* folder)
   - A visual studio solution which allows to run the console application and the unit tests (*SiriusQuality-PhenologyComponent.sln*)
   - A detailed documentation about both the calculation scheme and the equations of the component (*Documentation* folder)

2. How to use the component:

   The *SQ2_Phenology_component* can be added to a *BioMa* solution with the *CLIC* tool or be plugged to your model via a wrapper. Here the composition with *CLIC* will not be presented (see instead *BioMa* solution documentation). We will first make a quick overview of the component, then we will present the wrapper and finally we will explain how it is used via a console application.

   a. Overview of the component

      The component (*SiriusQuality-PhenologyComponent)* contains six folders:

      - The *domainClass* folder. Here can be found:
        - The getter and setter of the states (*PhenologyState.cs*)
        - The metadata on the states (*PhenologyStateVarInfo.cs*)
        - Two special classes which are not created with the BioMA tools: *Calendar.cs* where the object

calendar is defined to register the developmental stages ((*string*) Name of the stage, (*DateTime*) start date, (*double*) Thermal time at start); *Phase.cs* where the phase object, which is a decimal code for the growth phases, is declared

- The *strategies* folder where the simple strategies and the composite one (*phenology.cs*) can be found. The composite strategy allows to call sequentially the simple strategies via its *Estimate* function
- The *API* folder containing the classes for the Application Programming Interface
- An *XML* folder where the xml files used to generate strategies and domain classes with the BioMa tools can be found
- The *obj/Debug* folder containing the dll of the component after having built it and the BioMA dll which are mandatory for the project
- The *bin* folder for binaries

b. Phenology wrapper

The wrapper (*SiriusQuality-PhenologyConsole/ PhenologyWrapper.cs*) makes possible:

- The initial loading of the parameters
- The day by day valorization of the inputs
- The daily call of the component
- And the daily export of the outputs

For these purposes two *PhenologyState* objects are instantiated; one for states of the day before (*previousphenologyState* or *phenologystate1* in the component) and one for the current day (*phenologyState* or *phenologystate* in the component). These objects are used to valorize inputs and export outputs (via getter). In addition, an object of the composite class is instantiated (*phenologyComponent*). It is used to valorize the parameters and call the *Estimate* function of the composite.

The valorization of the parameters is done in the constructor via the *LoadParameters* function. The *EstimatePhenology* function can be called everywhere in the code. Its arguments are the values of the input for the current day. Three steps are necessary in the *EstimatePhenology* function:

- First the states of the day before are transferred from *phenologyState* to *previousPenologyState* via the *LoadPreviousStates* function
- Then the current day inputs are valorized
- Finally, the *Estimate* function of the composite class is called

The last function to be mentioned, *Init*, calls the initialization function of the composite class and valorize the parameters via the *LoadParameters* function. The initialization function of the composite class registers the Zadok 00 stage (sowing), valorize the potential final leaf number and initiate the state of the *CalculateShootNumber* simple strategy.

c. Console application

The *SiriusQuality-PhenologyConsole/Program.cs* class can be divided in five parts:

- Inputs that cannot be calculated are grouped in tables. The table counts correspond to the number of days in the simulation
- An object *PhenologyWrapper* is instantiated to be able to call the *phenologyEstimate* function and export outputs
- Initialization is done via the *Init* function of the wrapper
- Other inputs are calculated and the *EstimatePhenology* function of the wrapper is call each day of the simulation until the maturity is reached or the number of day exceeds the table counts of step 1. As in the universe of *SiriusQuality2* a while loop is used
- Outputs of the component are called from the wrapper and printed

3. List of the provided libraries

   Six libraries are mandatory to be able to run the component. Five of them are part of the BioMa framework and are loaded both in the component and in the console application projects, while *CRA.AgroManagement2014.dll*, *CRA.AgroManagement2014.Impacts.dll* and *CRA.AgroManagement2014.dll* are used for the management of agronomic events (irrigation, fertilization…), *CRA.Core.Preconditions.dll* is used for the test of parameter, input and output values and *CRA.ModelLayer.dll* is dedicated to the generation of domain class and strategies.

   When building the *SiriusQuality-PhenologyComponent* project a library called *SiriusQuality-PhenologyComponent.dll* is created. It is loaded in the console application project (and the corresponding include is done on top of the wrapper class).

   These six libraries have to be loaded in any project aiming at working with the phenology component.

4. List of inputs, intermediate variables, outputs and special parameters of the component:

Inputs		
Name in the code	units	Definition
DayLength	hour	Photoperiod
DeltaTT	°Cd	Daily thermal time seen by the plant (Physiological thermal time)
cumulTT	°Cd	Cumulative physiological thermal time (can be calculated day by day from DeltaTT)
GrainCumulTT	°Cd	Cumulative thermal Time for grain (may be calculated with DeltaTT for days after anthesis)
GAI	m²/m²	Total Green Area Index of the canopy
PAR	MJ/m²/d	Photosynthetically Active Radiations intercepted by the canopy
currentdate	month/day/year	Date of the day when phenology is estimated
IsLatestLeafInternodeLengthPotPositive	dimensionless	Integer to trigger the stem elongation growth stage. When it is 1 the stem starts to elongates. (set it always to 0 to ignore this stage)

Intermediate variables are used for calculation purposes only; they may be exported as output but it does not always make sense

Intermediates variables		
Name in the code	units	Definition
HasFlagLeafLiguleAppeare	-	Integer: 1 if the flag leaf has appeared, 0 if not (The lea number on main-stem has reached the final leaf number)
MinFinalNumber	leaf	Potential final leaf number. See Vernalization in */Documentation/SQ2_Phenology_component.pdf* for more details
hasLastPrimordiumAppeared	-	Integer: 1 the las primordium has appeared, 0 if not. When the last primordium has appeared the floral initiation starts
isMomentRegistredZC_39	-	Integer Zadok moment 39 (Flag leaf ligule just visible) is registered in the calendar
cumulTTFromZC_39	°Cd	Cumulative thermal time since the beginning of Zadok moment 39 (Flag leaf ligule just visible)
cumulTTFromZC_91	°Cd	Cumulative thermal time since the beginning of Zadok moment 91 (End of grain filling)
cumulTTFromZC_65	°Cd	Cumulative thermal time since the beginning of Zadok moment 91 (Anthesis)
currentZadokStage	-	Zadok stage of the day
hasZadokStageChanged	°Cd	Integer: 1 if the Zadok stage has changed during the day, 0 if not
Fixphyll	°Cd/leaf	Phyllochron parameter (P) corrected by sowing date effect. See Sowing date correction in */Documentation/SQ2_Phenology_component.pdf* for more details
ListTTShootTTWindowForPTQ	°Cd	List containing the day by day DeltaTT in the widow TTWindowForPTQ for the sliding average of the PTQ calculation
ListPARTTWindowForPTQ	MJ/m²/d	List containing the day by day PAR in the widow TTWindowForPTQ for the sliding average of the PTQ calculation
GAImax	m²/m²	Maximum Green Area Index from sowing to the current day

Outputs		
Name in the code	units	Definition
(Calendar) Calendar	-	An object where the developmental stage, the date when it is attained and the corresponding cumulative thermal time is register
LeafNumber	leaf	Decimal number of leaves on the main-stem equivalent to Haun stage
FinalLeafNumber	leaf	Number of leaves on the main-stem at maturity
(Phase) phase_.phaseValue	-	A decimal which encode the growth phase. See Table 1 in */Documentation/SQ2_Phenology_component.pdf* for more details
Phyllochron	°Cd/leaf	Phyllochron (Inverse of the leaf apparition rate)
VernaProg	-	Integer in [0-1] which represents the vernalization progress (0: start of vernalization, 1 end)
tilleringProfile	shoot/m²	List which stores the density of new tiller (main-stem +tillers) created at each time a new tiller appears
leafTillerNumberArray	shoot	List which stores the number of tillers (main-stem +tillers) for each leaf layer
CanopyShootNumber	shoot/m²	Shoot density (main-stem + tillers)
AverageShootNumberPerPlant	shoot	Average number of tillers (main-stem + tillers) per plant
TillerNumber	shoot	Number of tiller (main-stem + tillers)
PTQ	MJ/(°C.d.m²)	Photothermal Quotient See */Documentation/SQ2_Phenology_component.pdf* for more details

A list of the parameters dedicated to the calculations of the component is given in Table A2 of */Documentation/SQ2_Phenology_component.pdf*. In the present document we consider only the *"special parameters"* not described in the *SQ2_Phenology_component* documentation.

Special parameters		
Name in the code	units	Definition
TTWindowForPTQ	°Cd	Thermal Time window for the sliding average of the PTQ calculations
IgnoreGrainMaturation	-	Integer: 1 Zadok stage 92 (or phase 6: Maturity or dry grains) is ignored, 0 it is not
choosePhyllUse	-	Option to choose the phyllochron model: • "Default", segmented linear model with sowing date correction • "PTQ", modeled from Photothermal Quotient • "Test", segmented linear model without sowing date correction (it is possible to test a linear modeled phyllochron by setting parameters Pdecr an Pincr to 1.0) See */Documentation/SQ2_Phenology_component.pdf* for more details